Background Subtraction with Dirichlet Processes

Tom S.F. Haines and Tao Xiang

Electronic Engineering and Computer Science, Queen Mary, Uni. of London
{thaines,txiang}@eecs.qmul.ac.uk

Abstract. Background subtraction is an important first step for video analysis, where it is used to discover the objects of interest for further processing. Such an algorithm often consists of a background model and a regularisation scheme. The background model determines a per-pixel measure of if a pixel belongs to the background or the foreground, whilst the regularisation brings in information from adjacent pixels. A new method is presented that uses a Dirichlet process Gaussian mixture model to estimate a per-pixel background distribution, which is followed by probabilistic regularisation. Key advantages include inferring the per-pixel mode count, such that it accurately models dynamic backgrounds, and that it updates its model continuously in a principled way.

1 Introduction

Background subtraction can be defined as separating a video stream into the regions unique to a particular moment in time (the foreground), and the regions that are always present (the background). It is primarily used as an interest detector for higher level problems, such as automated surveillance, intelligent environments and motion analysis. The etymology of *background subtraction* derives from the oldest method, where a single static image of just the background is subtracted from the current frame, to generate a difference image. If the absolute difference exceeds a threshold the pixel in question is declared to belong to the foreground. Such an approach fails because the background is rarely static. Background variability has many underlying causes [1,2]:

Dynamic background, where objects such as trees blow in the wind, escalators move and traffic lights change colour.
Noise, as caused by the image capturing process. It can vary over the image due to photon noise and varying brightness.
Camouflage, where a foreground object looks very much like the background, e.g. a sniper in a ghillie suit.
Moved object, where the background changes, e.g. a car could be parked in the scene, and after sufficient time considered part of the background, only to later become foreground again when driven off.
Bootstrapping. As it is often not possible to get a frame with no foreground an algorithm should be capable of being initialised with foreground objects in the scene. It has to learn the correct background model over time.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part IV, LNCS 7575, pp. 97–111, 2012.

Illumination changes, both gradual, e.g. from the sun moving during the day, and rapid, such as from a light switch being toggled.
Shadows are cast by the foreground objects, but later processing is typically not interested in them.

The background subtraction field is gargantuan, and has many review papers [3,4,5,2]. Stauffer & Grimson [6] is one of the best known approaches - it uses a Gaussian mixture model (GMM) for a per-pixel density estimate (DE) followed by connected components for regularisation. This model improves on using a background plate because it can handle a *dynamic background* and *noise*, by using multimodal probability distributions. As it is continuously updated it can *bootstrap*. Its mixture model includes both foreground and background components - it classifies values based on their mixture component, which is assigned to the foreground or the background based on the assumption that the majority of the larger components belong to the background, with the remainder foreground. This assumption fails if objects hang around for very long, as they quickly dominate the distribution. The model is updated linearly using a fixed learning rate parameter - it is not very good with the *moved object* problem. Connected components converts the intermediate foreground mask into regions via pixel adjacency, and culls all regions below a certain size, to remove spurious detections. This approach to noise handling combined with its somewhat primitive density estimation method undermines *camouflage* handling, as it often thinks it is noise, and also prevents it from tracking small objects. No capacity exists for it to handle *illumination changes* or *shadows*. The above can be divided into 4 parts - the model, updating the model, how pixels are classified, and regularisation; alternate approaches for each will now be considered in turn.

The Model: Alternative DE methods exist, including different GMM implementations [7] and kernel density estimate (KDE) methods, either using Gaussian kernels [8,9] or step kernels [10,7]. Histograms have also been used [11], and alternatives to DE include models that predict the next value [1], use neural networks [12], or hidden Markov models [13]. An improved background model should result in better performance regarding *dynamic background*, *noise* and *camouflage*. This is due to better handling of underfitting and/or overfitting, which improves generalisation to the data stream. Whilst better than Stauffer & Grimson [6] the above methods still suffer from over/under-fitting. KDE and histogram methods are particularly vulnerable, as they implicitly assume a constant density by using fixed size kernels/bins. GMM methods should do better, but the heuristics required for online learning, particularly regarding the creation of new components, can result in local minima in the optimisation, which is just as problematic.

Our Approach: We present an approach that uses a Dirichlet process Gaussian mixture model (DP-GMM) [14] for per-pixel density estimation. This is a non-parametric Bayesian method [15] that automatically estimates the number of mixture components required to model the pixels background colour distribution. Consequentially it correctly handles multi-modal *dynamic backgrounds* with regular colour/luminance changes, such as trees waving in the wind. As a fully Bayesian model over-fitting is avoided, improving robustness to *noise* and classifying pixels precisly, which helps to distinguish noise from *camouflage*. He et al. [16] recently also used DP-GMMs for background subtraction, in a block-based method. They failed to leverage the potential advantages however (Discussed below), and used computationally unmanageable methods - despite their efforts poor results were obtained.

Model Update: Most methods use a constant learning rate to update the model, but some use adaptive heuristics [7,17], whilst others are history based [1,16], and build a model from the last n frames directly. Adapting the learning rate affects the *moved object* issue - if it is too fast then stationary objects become part of the background too quickly, if it is too slow it takes too long to recover from changes to the background. Adaptation aims to adjust the rate depending on what is happening. Continuously learning the model is required to handle the *bootstrapping* issue.

Our Approach: Using a DP-GMM allows us to introduce a novel model update concept that lets old information degrade in a principled way. One side effect of this and the use of Gibbs sampling is that no history has to be kept [1,16], avoiding the need to store and process hundreds of frames. It works by capping the confidence of the model, i.e. limiting how certain it can be about the shape of the background distribution. This allows a stationary object to remain part of the foreground for a very long time, as it takes a lot of information for the new component to obtain the confidence of pre-existing components, but when an object moves on and the background changes to a component it has seen before, even if a while ago, it can use that component immediately. Updating the components for gradual background changes continues to happen quickly, making sure the model is never left behind. Confidence capping works because non-parameteric Bayesian models, such as DP-GMMs, have a rigorous concept of a new mixture component forming - parametric models [6,7] have to use heuristics to simulate this, whilst KDE based approaches are not compatible [8,9,10,7] as they lack a measure of confidence.

Pixel Classification: The use of a single density estimate that includes both foreground (fg) and background (bg), as done by Stauffer & Grimson [6] is somewhat unusual - most methods stick to separate models and apply Bayes rule [11], with the foreground model set to be the uniform distribution as it is unknown. **Our approach:** We follow this convention, which results in a probability of being bg or fg, rather than a hard classification, which is passed through to the regularisation step. Instead of using Bayes rule some works use a threshold [8]. Attempts at learning a foreground model also exist [9], and some models generate a binary classification directly [12].

Regularisation: Some approaches have no regularisation step [18], others have information sharing between adjacent pixels [12] but no explicit regularisation. Techniques such as eroding then dilating are common [2], and more advanced techniques have, for instance, tried to match pixels against neighbouring pixels, to compensate for background motion [8]. When dealing with a probabilistic fg/bg assignment probabilistic methods should be used, such as the use of Markov random fields (MRF) by Migdal & Grimson [19] and Sheikh & Shah [9].

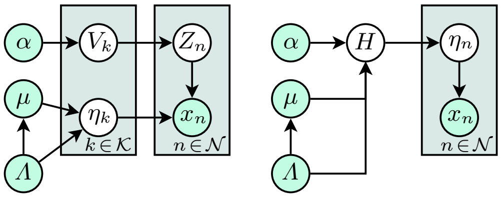
Our Approach: We use the same method - the pixels all have a random variable which can take on one of two labels, fg or bg. The data term is provided by the model whilst pairwise potentials indicate that adjacent pixels should share the same label. Differences exist - previous works use Gibbs sampling [19] and graph cuts [9], whilst we choose belief propagation [20], as run time can be capped; also we use an edge preserving cost between pixels, rather than a constant cost, which proves to be beneficial with high levels of noise. Cohen [21] has also used a Markov random field, but to generate a background image by selecting pixels from a sequence of frames, rather than for regularisation.

2 Methodology

2.1 Per-Pixel Background Model

Each pixel has a density estimate constructed for it, to model $P(x|\text{bg})$ where x is the value of the pixel. The Dirichlet process Gaussian mixture model (DP-GMM) [14] is used. It can be viewed as the Dirichlet distribution extended to an infinite number of components, which allows it to learn the true number of mixtures from the data. For each pixel a stream of values arrives, one with each frame - the model has to be continuously updated with incremental learning.

Figure 1a represents the DP-GMM graphically using the *stick breaking* construction; it can be split into 3 columns - on the left the priors, in the middle the entities representing the Dirichlet process (DP) and on the right the data for which a density estimate is being constructed. This last column contains the feature vectors (pixel colours) to which the model is being fitted, x_n, which come from all previous frames, $n \in \mathcal{N}$. It is a generative model - each sample comes from a specific mixture component, indexed by $Z_n \in \mathcal{K}$, which consists of its probability of being selected, V_k and the Gaussian distribution from which the value was drawn, η_k. The conjugate prior, consisting of μ, a Gaussian over its mean, and Λ, a Wishart distribution over its inverse covariance matrix, is applied to all η_k. So far this is just a mixture model; the interesting part is that $\mathcal{K}$, the set of mixture components, is infinite. Conceptually the stick breaking construction is very simple - we have a stick of length 1, representing the entire probability mass, which we keep breaking into two parts. Each time it is broken one of the parts becomes the probability mass for a mixture component - a value of V_k, whilst the other is kept for the next break. This continues forever. α is the concentration parameter, which controls how the stick is broken - a low value puts most of the probability mass in a few mixture components, whilst a high value spreads it out over many. Orthogonal to the stick length each stick is associated with a draw, η_k, from the DP's base measure, which is the already mentioned conjugate prior over the Gaussian.



(a) Stick breaking (b) Chinese restaurant

Fig. 1. Two versions of the DP-GMM graphical model

Whilst the stick breaking construction offers a clean explanation of the model the Chinese restaurant process (CRP) is used for the implementation[1]. This is the model with the middle column of Figure 1a integrated out, to give Figure 1b. It is named by analogy. Specifically, each sample is represented by a customer, which turns up and sits at a table in a Chinese restaurant. Tables represent the mixture components, and a customer chooses either to sit at a table where customers are already sitting, with probability proportional to the number of customers at that table, or to sit at a new table, with probability proportional to α. At each table (component) only one dish is consumed, which is chosen from the menu (base measure) by the first customer to sit at that table. Integrating out the draw from the DP leads to better convergence, but more importantly replaces the infinite set of sticks with a computationally tractable finite set of tables.

Each pixel has its own density estimate, updated with each new frame. Updating proceeds by first calculating the probability of the current pixel value, x, given the current background model, then updating the model with x, weighted by the calculated probability - these steps will now be detailed.

Mixture Components: The per-pixel model is a set of weighted mixture components, such that the weights sum to 1, of Gaussian distributions. It is integrated out however, using the Chinese restaurant process for the mixture weights and the conjugate prior for the Gaussians. Whilst the literature [23] already details this second part it is included for completeness. $x \in [0,1]^3$ represents the pixels colour, and independence is assumed between the components for reasons of speed. This simplifies the Wishart prior to a gamma prior for each channel i, such that

$$\sigma_i^{-2} \sim \Gamma\left(\frac{n_{i,0}}{2}, \frac{\sigma_{i,0}^2}{2}\right), \qquad \mu_i|\sigma_i^2 \sim \mathcal{N}\left(\mu_{i,0}, \frac{\sigma_i^2}{k_{i,0}}\right), \qquad x_i \sim \mathcal{N}(\mu_i, \sigma_i^2), \tag{1}$$

where $\mathcal{N}(\mu, \sigma^2)$ represents the normal distribution and $\Gamma(\alpha, \beta)$ the gamma distribution. The parameters $n_{i,0}$ and $\sigma_{i,0}$, $i \in \{0,1,2\}$, are the Λ prior from the graphical model, whilst $\mu_{i,0}$ and $k_{i,0}$ are the μ prior.

[1] Variational methods [22] offer one approach to using the stick breaking construction directly. This is impractical however as historic pixel values would need to be kept.

Evidence, x, is provided incrementally, one sample at a time, which will be weighted, w. The model is then updated from having m samples to $m+1$ samples using

$$n_{i,m+1} = n_{i,m} + w, \qquad k_{i,m+1} = k_{i,m} + w,$$
$$\mu_{i,m+1} = \frac{k_{i,m}\mu_{i,m} + wx_i}{k_{i,m} + w}, \qquad \sigma^2_{i,m+1} = \sigma^2_{i,m} + \frac{k_{i,m}w}{k_{i,m} + w}(x_i - \mu_{i,m})^2. \tag{2}$$

Note that $n_{i,m}$ and $k_{i,m}$ have the same update, so one value can be stored to cover both, for all i. Given the above parameters, updated with the available evidence, a Gaussian may be drawn, to sample the probability of a colour being drawn from this mixture component. Instead of drawing it the Gaussian is integrated out, to give

$$x_i \sim \mathcal{T}\left(n_{i,m}, \mu_{i,m}, \frac{k_{i,m} + 1}{k_{i,m}n_{i,m}}\sigma^2_{i,m}\right), \tag{3}$$

where $\mathcal{T}(v, \mu, \sigma^2)$ denotes the three parameter student-t.

Background Probability: To calculate the probability of a pixel, $x \in [0,1]^3$, belonging to the background (bg) model the Chinese restaurant process is used. The probability of x given component (table) $t \in T$ is

$$P(x|t, \text{bg}) = \frac{s_t}{\sum_{i \in T} s_i} P(x|n_t, k_t, \mu_t, \sigma^2_t), \tag{4}$$
$$P(x|n_t, k_t, \mu_t, \sigma^2_t) = \prod_{i \in \{0,1,2\}} \mathcal{T}\left(x_i | n_{t,i}, \mu_{t,i}, \frac{k_{t,i} + 1}{k_{t,i}n_{t,i}}\sigma^2_{t,i}\right), \tag{5}$$

where s_t is the number of samples assigned to component t, and n_t, μ_t, k_t and σ_t are the parameters of the prior updated with the samples currently assigned to the component. By assuming the existence of a dummy component, $t = \text{new} \in T$, that represents creating a new component (sitting at a new table) with $s_{\text{new}} = \alpha$ this is the Chinese restaurant process. The student-t parameters for this dummy component are the prior without update. Finally, the mixture components can be summed out

$$P(x|\text{bg}) = \sum_{t \in T} P(x|t, \text{bg}). \tag{6}$$

The goal is to calculate $P(\text{bg}|x)$, not $P(x|\text{bg})$, hence Bayes rule is applied,

$$P(\text{bg}|x) = \frac{P(x|\text{bg})P(z\text{bg})}{P(x|\text{bg}) + P(x|\text{fg})}, \tag{7}$$

noting that pixels can only belong to the background or the foreground (fg), hence the denominator. $P(x|\text{bg})$ is given above, leaving $P(\text{bg})$ and $P(x|\text{fg})$. $P(\text{bg})$ is an implicit threshold on what is considered background and what is considered foreground, and is hence considered to be a parameter[2]. $P(x|\text{fg})$ is unknown and hard to estimate, so the uniform distribution is used, which is a value of 1, as the volume of the colour space is 1 (See subsection 2.3).

[2] Though it is simply set to 0.5 for the majority of the experiments.

Model Update: To update the model at each pixel the current value is assigned to a mixture component, which is then updated - s_t is increased and the posterior for the Gaussian updated with the new evidence. Assignment is done probabilistically, using the term for each component from Equation 4, including the option of a new mixture component. This is equivalent to Gibbs sampling the density estimate, except we only sample each value once on arrival. Updates are weighted by their probability of belonging to the background (Equation 7). Sampling each value just once is not an issue, as the continuous stream of data means the model soon converges.

A learning rate, as found in methods such as Stauffer & Grimson [6], is not used; instead, unique to a DP-GMM, the confidence of the model is capped. This can be interpreted as an adaptive update [7,17], but it is both principled and very effective. In effect we are building a density estimate with the ability to selectively forget, allowing newer data to take over when the background changes. It works by capping how high s_t can go, noting that s_t is tied to n_t and k_t, so they also need to be adjusted. When this cap is exceeded a multiplier is applied to all s_t, scaling the highest s_t down to the cap. Note that σ_t^2 is dependent on k_t, as it includes k_t as a multiplier - to avoid an update σ_t^2/k_t is stored instead. The effectiveness is such that it can learn the initial model with less than a second of data yet objects can remain still for many minutes before being merged into the background, without this impeding the ability of the model to update as the background changes. Finally, given an infinite number of frames the number of mixture components goes to infinity, so the number is capped. When a new component is created the existing component with the lowest s_t is replaced.

2.2 Probabilistic Regularisation

The per-pixel background model ignores information from a pixels neighbourhood, leaving it susceptible to noise and camouflage. To resolve this a Markov random field is constructed, with a node for each pixel, connected using a 4-way neighbourhood. It is a binary labelling problem, where each pixel either belongs to the foreground or the background. The task is to select the most probable solution, where the probability can be broken up into two terms. Firstly, each pixel has a probability of belonging to the background or foreground, directly obtained from the model as $P(\text{bg}|x)$ and $1 - P(\text{bg}|x)$, respectively. Secondly, there is a similarity term, which indicates that adjacent pixels are likely to have the same assignment,

$$P(l_a = l_b) = \frac{h}{h + m * d(a, b)}, \tag{8}$$

where l_x is the label of pixel x, h is the half life, i.e. the distance at which the probability becomes 0.5 and $d(a, b)$ is the Euclidean distance between the two pixels. m is typically 1, but is decreased if a pixel is sufficiently far from its neighbours that none provides a $P(l(a) = l(b))$ value above a threshold. This encourages a pixel to have a similar label to its neighbours, which filters out noise. Various methods can be considered for solving this model. Graph cuts [24] would give the MAP solution, however we use belief propagation instead [20], as it runs in constant time given an iteration cap, which is important for a real time implementation; it is also more amenable to a GPU implementation.

2.3 Further Details

The core details have now been given, but other pertinent details remain.

The Prior: The background model includes a prior on the Gaussian associated with each mixture component. Instead of treating this as a parameter to be set it is calculated from the data. Specifically, the mean and standard deviation (SD) of the prior are matched with the mean and SD of the pixels in the current frame,

$$n_{i,0} = k_{i,0} = 1, \qquad \mu_{i,0} = \frac{1}{|F|}\sum_{x \in F} x_i, \qquad \sigma^2_{i,0} = \frac{1}{|F|}\sum_{x \in F}(x_i - \mu_{i,0})^2, \tag{9}$$

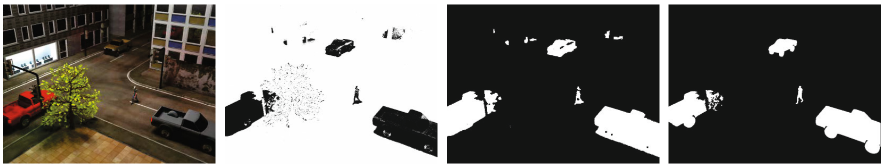
where F is the set of pixels in the current frame. To change the prior between frames the posterior parameters must not be stored directly. Instead offsets from the prior are stored, which are then adjusted after each update such that the model is equivalent. The purpose then is to update the distribution that mixture components return to as they lose influence, to keep that in line with the current lighting level.

Lighting Change: The above helps by updating the prior, but it does nothing to update the evidence. To update the evidence a multiplicative model is used, whereby the lighting change between frames is estimated as a multiplier, then the entire model is updated by multiplying the means, $\mu_{i,m}$, of the components accordingly. Light level change is estimated as in Loy et al. [25]. This takes every pixel in the frame and divides its value by the same pixel in the previous frame, as an estimate of the lighting change. The mode of these estimates is then found using mean shift [26], which is robust to the many outliers.

Colour Model: A simple method for filtering out shadows is to separate the luminance and chromaticity, and then ignore the luminance, as demonstrated by Elgammal et al. [8]. This tends to ignore too much information; instead the novel step is taken of reducing the importance of luminance. In doing so luminance is moved to a separate channel; due to the DE assuming independence between components this is advantageous, as luminance variation tends to be higher than chromatic variation. To do this a parametrised colour model is designed. First the r, g, b colour space is rotated so luminance is on its own axis

$$\begin{pmatrix} l \\ m \\ n \end{pmatrix} = \begin{pmatrix} \sqrt{3} & \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{2} & -\sqrt{2} \\ -2\sqrt{6} & \sqrt{6} & \sqrt{6} \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix}, \tag{10}$$

then chromaticity is extracted



(a) Input video frame. (b) $P(\text{bg}|\text{model})$ - output of the DP-GMM for each pixel. (c) Foreground mask generated by the presented approach. (d) Ground truth foreground mask.

Fig. 2. Frame 545 from the bootstrap sequence

Table 1. Brief summaries of all the algorithms compared against

Barnich [10]	KDE with a spherical kernel. Uses a stochastic history.
Collins [27]	Hybrid frame differencing / background model.
Culibrk [28]	Neural network variant of Gaussian KDE.
Kim [18]	'Codebook' based; almost KDE with a cuboid kernel.
Li 1 [11]	Histogram based, includes co-occurrence statistics. Lots of heuristics.
Li 2 [29]	Refinement of the above.
Maddalena [12]	Uses a self organising map, passes information between pixels.
Stauffer [6]	Classic GMM approach. Assigns mixture components to bg/fg.
Toyama [1]	History based, with region growing. Has explicit light switch detection.
Wren [30]	Incremental spatio-colourmetric clustering (tracking) with change detection.
Zivkovic [7]	Refinement of Stauffer [6]. Has an adaptive learning rate.

$$l' = 0.7176\, l, \qquad \begin{pmatrix} m' \\ n' \end{pmatrix} = \frac{0.7176}{\max(l, f)} \begin{pmatrix} m \\ n \end{pmatrix}, \tag{11}$$

where 0.7176 is the constant required to maintain a unit colour space volume[3]. To obtain chromaticity the division should be by l rather than $\max(l, f)$, but this results in a divide by zero. Assuming the existence of noise when measuring r, g, b the division by l means the variance of m' and n' is proportional to $\frac{1}{l^2}$. To limit variance as well as extract chromaticity, we have two competing goals - the use of $\max(l, f)$ introduces f, a threshold on luminance below which capping variance takes priority. Given this colour space it is then parametrised by r, which scales the luminance to reduce its importance against chromaticity

$$[l, m, n]_r = [r^{\frac{2}{3}} l', r^{-\frac{1}{3}} m', r^{-\frac{1}{3}} n']. \tag{12}$$

The volume of the colour space has again been held at 1. Robustness to shadows is obtained by setting r to a low value, as this reduces the importance of brightness changes.

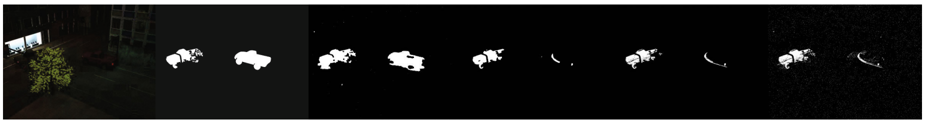
3 Experiments

Three sets of results are demonstrated - the synthetic test of Brutzer et al. [2] and two real world tests - *wallflower* from Toyama et al. [1] and *star* from Li et al. [29].

[3] The post processor assumes a uniform distribution over colour, and hence needs to know the volume. Note that this constant does not account for f, but then it makes very little difference to the volume.

Table 2. Synthetic experimental results - f-measures for each of the 9 challenges. The results for other algorithms were obtained from the website associated with Brutzer et al. [2], though algorithms that never got a top score in the original chart have been omitted. The numbers in brackets indicate which is the best, second best etc. The mean column gives the average for all tests - the presented approach is 27% higher than its nearest competitor.

method	basic	dynamic background	bootstrap	darkening	light switch	noisy night	camouflage	no camouflage	h.264, 40kbps	*mean*
Stauffer [6]	.800 (3)	.704 (5)	.642 (5)	.404 (7)	.217 (6)	.194 (6)	.802 (4)	.826 (4)	.761 (6)	.594 (7)
Li 1 [11]	.766 (5)	.641 (6)	.678 (4)	.704 (3)	.316 (3)	.047 (7)	.768 (6)	.803 (6)	.773 (4)	.611 (5)
Zivkovic [7]	.768 (4)	.704 (5)	.632 (6)	.620 (6)	.300 (4)	.321 (3)	.820 (3)	.829 (3)	.748 (7)	.638 (3)
Maddalena [12]	.766 (5)	.715 (3)	.495 (7)	.663 (5)	.213 (7)	.263 (5)	.793 (5)	.811 (5)	.772 (5)	.610 (6)
Barnich [10]	.761 (6)	.711 (4)	.685 (3)	.678 (4)	.268 (5)	.271 (4)	.741 (7)	.799 (7)	.774 (3)	.632 (4)
DP, no post	.836 (2)	.827 (2)	.717 (2)	.736 (2)	.499 (2)	.346 (2)	.848 (2)	.851 (2)	.781 (2)	.715 (2)
DP	**.853** (1)	**.853** (1)	**.796** (1)	**.861** (1)	**.603** (1)	**.788** (1)	**.864** (1)	**.867** (1)	**.827** (1)	**.812** (1)
DP, con com	.855	.872	.722	.818	.500	.393	.847	.851	.838	.744
DP, rgb	.850	.859	.783	.807	.445	.334	.852	.857	.848	.737



(a) Input (b) Ground truth (c) DP (d) DP, rgb (e) DP, con com (f) DP, no post

Fig. 3. Frame 990 from the noisy night sequence

Brutzer et al. [2] introduced a synthetic evaluation procedure for background subtraction algorithms, consisting of a 3D rendering of a junction, traversed by both cars and people - see Figure 2. Despite being synthetic it simulates, fairly accurately, 9 real world problems, and has the advantage of ground truth for all frames. The f-measure is reported for the various approaches in Table 2, and is defined as the harmonic mean of the recall and precision. Table 1 summarises all the algorithms compared against during all the experiments. For this test we used one set of parameters for all problems, rather than tuning per problem[4]. As can be seen, the presented approach takes the top position for all scenarios, being on average 27% better than its nearest competitor, and in doing so demonstrates that it is not sensitive to the parameters chosen. The algorithm without regularisation is also included in the chart[5] - in all cases a lack of regularisation does not undermine its significant lead over the competition, demonstrating that the DP-GMM is doing most of the work, but that regularisation always improves the score, on average by 13%. It can be noted that the largest performance gaps between regularisation being off and being on appears for the nosiest inputs, e.g. noisy night, light switch, darkening and h264. These are the kinds of problems encountered in surveillance applications. As a further point of comparison *DP, con com* is included, where the post-processing has been swapped for the connected components method of Stauffer & Grimson [6]. Interestingly for the simpler problems it does very well, sometimes better than the presented method,

[4] The original paper tuned one parameter per problem - we are at a disadvantage.

[5] The other algorithms on the chart have had their post-processing removed, so it can be argued that this is the fairer comparison to make, though Brutzer et al. [2] define post-processing such that our regularisation method is allowed.

(a) moved object (b) time of day (c) light switch (d) waving trees (e) camouflage (f) bootstrap (g) fg aperture

Fig. 4. Results for the *wallflower* dataset - on the top row is the image, on the second row the ground truth and on the third row the output of the presented algorithm. Toyama et al. [1] provide the outputs for other algorithms.

Table 3. Results for the *wallflower* dataset [1], given as the number of pixels that have been assigned the wrong class. Again, weaker algorithms have been culled from the original, though the positions continue to account for the missing methods. On average the presented approach makes 33% less mistakes than its nearest competitor.

method	moved object	time of day	light switch	waving trees	camouflage	bootstrap	foreground aperture	*mean*
Frame difference	**0** (1)	1358 (12)	2565 (3)	6789 (16)	10070 (12)	2175 (4)	4354 (9)	3902 (8)
Mean + threshold	**0** (1)	2593 (15)	16232 (11)	3285 (13)	1832 (3)	3236 (9)	2818 (5)	4285 (9)
Mixture of Gaussians	**0** (1)	1028 (10)	15802 (8)	1664 (8)	3496 (6)	2091 (3)	2972 (6)	3865 (7)
Block correlation	1200 (11)	1165 (11)	3802 (4)	3771 (15)	6670 (11)	2673 (8)	2402 (4)	3098 (5)
Eigen-background	1065 (10)	895 (7)	1324 (2)	3084 (12)	1898 (4)	6433 (11)	2978 (7)	2525 (3)
Toyama [1]	**0** (1)	986 (8)	**1322** (1)	2876 (11)	2935 (5)	2390 (6)	**969** (1)	1640 (2)
Maddalena [12]		453 (2)		293 (3)				
Wren [30]		654 (6)		298 (4)				
Collins [27]		653 (5)		430 (6)				
Kim [18]		492 (3)		353 (5)				
DP	**0** (1)	596 (4)	15071 (6)	265 (2)	1735 (2)	1497 (2)	1673 (3)	2977 (4)
DP, tuned	**0** (1)	**330** (1)	3945 (5)	**184** (1)	**384** (1)	**1236** (1)	1569 (2)	**1093** (1)

but when it comes to the trickier scenarios the presented is clearly better. To justify the use of the parametrised colour model *DP, rgb* shows the full model run using *rgb* instead of ours. The consequences are similar to those for connected components. Figure 3 shows all the variants for a frame from noisy night. It can be observed that the main advantage of the presented post processor is its ability to go from a weak detection that falls below the implicit threshold to a complete object, using both the colour and model uncertainty of the moving object.

The frame shown in Figure 2 has been chosen to demonstrate two weaknesses with the algorithm. Specifically, its robustness to shadows is not very effective - whilst this can be improved by reducing the importance of luminance in the colour space this has the effect of reducing its overall ability to distinguish between colours, and damages performance elsewhere. The second issue can be seen in the small blobs at the top of the image - they are actually the reflections of objects in the scene. Using a DP-GMM allows it to learn a very precise model,

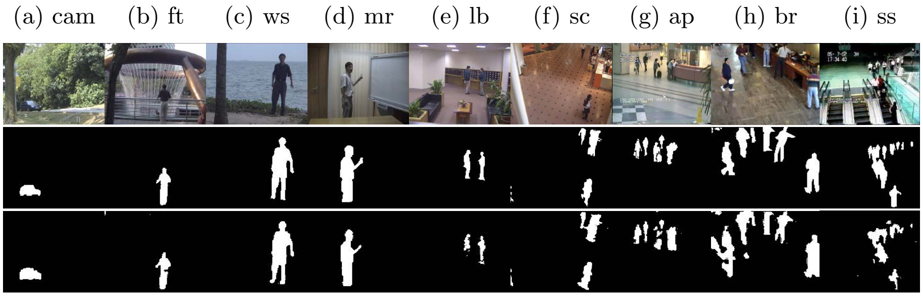


Fig. 5. Results for the *star* dataset - with the same frames as Culibrk et al. [28] and Maddalena & Petrosino [12], for a qualitative comparison. Layout is identical to Figure 4. The videos are named using abbreviations of their locations.

Table 4. Results for the *star* dataset [29,12]; refer to Figure 5 for exemplar frames, noting that *lb* has abrupt lighting changes. The average improvement of *DP, tuned* over its nearest competitor is 4%.

method	cam	ft	ws	mr	lb	sc	ap	br	ss	*mean*
Li 2 [29]	.1596 (5)	.0999 (6)	.0667 (6)	.1841 (6)	.1554 (6)	.5209 (6)	.1135 (6)	.3079 (6)	.1294 (6)	.1930 (6)
Stauffer [6]	.0757 (6)	.6854 (3)	.7948 (4)	.7580 (4)	.6519 (2)	.5363 (5)	.3335 (5)	.3838 (5)	.1388 (5)	.4842 (5)
Culibrk [28]	.5256 (4)	.4636 (5)	.7540 (5)	.7368 (5)	.6276 (4)	.5696 (4)	.3923 (4)	.4779 (4)	.4928 (4)	.5600 (4)
Maddalena [12]	.6960 (3)	.6554 (4)	.8247 (3)	.8178 (3)	.6489 (3)	.6677 (2)	**.5943** (1)	.6019 (3)	**.5770** (1)	.6760 (2)
DP	.7567 (2)	.7049 (2)	.9090 (2)	.8203 (2)	.5794 (5)	.6522 (3)	.5484 (3)	.6024 (2)	.5055 (3)	.6754 (3)
DP, tuned	**.7624** (1)	**.7265** (1)	**.9134** (1)	**.8371** (1)	**.6665** (1)	**.6721** (1)	.5663 (2)	**.6273** (1)	.5269 (2)	**.6998** (1)

so much so that it can detect the slight deviation caused by a reflection, when it would be preferable to ignore it. Further processing could avoid this.

Despite its low resolution (160×120) the *wallflower* [1] data set is one of the few real world options for background subtraction testing. It tests one frame only for each problem, by counting the number of mistakes made[6]; testing on a single frame is hardly ideal. There are seven tests, given in Figure 4 for a qualitative evaluation. Quantitative results are given in Table 3. Previously published results have been tuned for each problem, so we do the same in the *DP, tuned* row, but results using a single set of parameters are again shown, in the *DP* row, to demonstrate its high degree of robustness to parameter selection. For 5 of the 7 tests the method takes 1$^{\text{st}}$, albeit shared for the moved object problem.

On foreground aperture it takes 2$^{\text{nd}}$, beaten by the Toyama [1] algorithm. This shot consists of a sleeping person waking up, at which point they are expected to transition from background to foreground. They are wearing black and do not entirely move from their resting spot, so the algorithm continues to think they are background in that area. The regularisation helps to shrink this spot, but the area remains. It fails with the light switch test, which is interesting as no issue occurs with the synthetic equivalent. For the presented approach lighting correction consists of estimating a single multiplicative constant - this works outdoors where it is a reasonable model of the sun, but indoors where light bounces around and has

[6] For the purpose of comparison the error metrics used by previous papers [1] have been used.

a highly non-linear effect on the scene it fails. It is therefore not surprising that the synthetic approach, which simulates a sun, works, whilst the indoor approach, which includes light coming through a door and the glow from a computer monitor, fails. Examining the output in Figure 4 it can be noted that it has not failed entirely - the test frame is only the 13^{th} frame after the light has been switched on, and the algorithm is still updating its model after the change.

Finally, the *star* evaluation [29] is presented, which is very similar to the *wallflower* set - a video sequence is shared. The sequences are generally much harder though, due to text overlays, systemic noise and some camera shake, and fewer algorithms have been run on this set. It has a better testing procedure, as it provides multiple test frames per problem, with performance measured using the average similarity score for all test frames, where similarity $= \text{tp}/(\text{tp} + \text{fn} + \text{fp})$. The presented approach[7] takes 1^{st} 7 times out of 9, beaten twice by Maddalena et al. [12]. Its two weak results can probably be attributed to camera shake, as the presented has no robustness to shaking, whilst Maddalena et al. [12] does, due to model sharing between adjacent pixels. The light switch test in this data set does not trip it up this time - the library where it occurs has a high ceiling and diffuse lighting, making multiplicative lighting much more reasonable. Complex dynamic backgrounds clearly demonstrate the strength of a DP-GMM, as evidenced by its 3 largest improvements (*cam*, *ft* and *ws*).

Using a DP-GMM is computationally demanding - the implementation obtains 25 frames per second with 160×120, and is $O(n)$ where $n = wh$ is the number of pixels[8]. This is not a major concern, as real time performance on high resolution input could be obtained using a massively parallel GPU implementation. Indeed, an incomplete effort at this has already increased the speed by a factor of 5, making 320×240 real time.

4 Conclusions

This work represents the cutting edge background subtraction method[9]. It takes the basic concept of the seminal work of Stauffer & Grimson [6] and applies up to date methods in a mathematically rigorous way. The key advantage is in using DP-GMMs, which handle new mixture components forming as more information becomes available, and build highly discriminative models. Using a confidence cap handles the dynamics of a scene much better than a heuristic approach to model updates. Despite its thorough theoretical basis implementation remains relatively simple[10]. Certain improvements can be considered. Combining information between pixels only as a regularisation step does not fully exploit the information available, and so a rigorous method of spatial information

[7] As for *wallflower* we tune per-problem, as the competition has done the same; results for a single set of parameters are again presented.
[8] Run on a single core of an Intel i7 2.67Ghz.
[9] Code is available from `http://www.thaines.com`
[10] 186 lines of C for the DP model and 239 lines for the post-processing.

transmission would be desirable. This would be particularly helpful when handling mild camera shake. Sudden complex lighting changes are not handled, which means it fails to handle some indoor lighting changes.

References

1. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practise of background maintenance. In: ICCV, pp. 255–261 (1999)
2. Brutzer, S., Hferlin, B., Heidemann, G.: Evaluation of background subtraction techniques for video surveillance. In: CVPR (2011)
3. Cheung, S.C.S., Kamath, C.: Robust techniques for background subtraction in urban traffic video. In: VCIP, vol. 5308, pp. 881–892 (2004)
4. Karaman, M., Goldmann, L., Yu, D., Sikora, T.: Comparison of static background segmentation methods. In: VCIP, vol. 5960, pp. 2140–2151 (2005)
5. Herrero, S., Bescós, J.: Background Subtraction Techniques: Systematic Evaluation and Comparative Analysis. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 33–42. Springer, Heidelberg (2009)
6. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR, vol. 2, pp. 637–663 (1999)
7. Zivkovica, Z., Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters, 773–780 (2006)
8. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Frame-rate Workshop, pp. 751–767 (2000)
9. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. PAMI 27(11), 1778–1792 (2005)
10. Barnich, O., Droogenbroeck, M.V.: Vibe: A powerful random technique to estimate the background in video sequences. In: Acoustics, Speech and Signal Processing, pp. 945–948 (2009)
11. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Foreground object detection from videos containing complex background. In: Proc. Multimedia, pp. 2–10 (2003)
12. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Tran. IP 17(7), 1168–1177 (2008)
13. Kato, J., Watanabe, T., Joga, S., Rittscher, J., Blake, A.: An hmm-based segmentation method for traffic monitoring movies. PAMI 24(9), 1291–1296 (2002)
14. Escobar, M.D., West, M.: Bayesian density estimation and inference using mixtures. J. American Statistical Association 90(430), 577–588 (1995)
15. Teh, Y.W., Jordan, M.I.: Hierarchical Bayesian Nonparametric Models with Applications. In: Bayesian Nonparametrics. Cambridge University Press (2010)
16. He, Y., Wang, D., Zhu, M.: Background subtraction based on nonparametric bayesian estimation. In: Int. Conf. Digital Image Processing (2011)
17. Lee, D.S.: Effective gaussian mixture learning for video background subtraction. PAMI 27(5), 827–832 (2005)
18. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Background modeling and subtraction by codebook construction. In: ICIP, vol. 5, pp. 3061–3064 (2004)
19. Migdal, J., Grimson, W.E.L.: Background subtraction using markov thresholds. In: Workshop on Motion and Video Computing, pp. 58–65 (2005)
20. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. In: CVPR, vol. 70(1), pp. 41–54 (2004)

21. Cohen, S.: Background estimation as a labeling problem. In: ICCV, pp. 1034–1041 (2005)
22. Blei, D.M., Jordan, M.I.: Variational inference for dirichlet process mixtures. Bayesian Analysis, 121–144 (2005)
23. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis. Chapman & Hall (2004)
24. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)
25. Loy, C.C., Xiang, T., Gong, S.: Time-delayed correlation analysis for multi-camera activity understanding. IJCV 90(1), 106–129 (2010)
26. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI 24(5), 603–619 (2002)
27. Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., Hasegawa, O., Burt, P., Wixson, L.: A system for video surveillance and monitoring. Technical report, CMU (2000)
28. Culibrk, D., Marques, O., Socek, D., Kalva, H., Furht, B.: Neural network approach to background modeling for video object segmentation. Neural Networks 18(6), 1614–1627 (2007)
29. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE Tran. IP 13(11), 1459–1472 (2004)
30. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. PAMI 19(7), 780–785 (1997)